(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 June 2001 (07.06.2001)

PCT

(10) International Publication Number
WO 01/41002 A1

(51) International Patent Classification[7]: G06F 17/30

(21) International Application Number: PCT/US00/32815

(22) International Filing Date: 1 December 2000 (01.12.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/452,849 2 December 1999 (02.12.1999) US

(71) Applicant: LOCKHEED MARTIN CORPORATION [US/US]; 6801 Rockledge Drive, Bethesda, MD 20817-1836 (US).

(72) Inventors: DI DOMIZIO, Virginia, Ann; 25 Latham Village Lane, #6, Latham, NY 12110 (US). DIXON, Walter, Vincent, III; 3641 Lake Road, Delanson, NY 12053 (US). EPTER, Scott, D.; 18 Cozine Avenue, Rhinebeck, NY 12572 (US). HOEBEL, Louis, John; 33 Forest Road, Burnt Hills, NY 12027 (US). OKSOY, Osman, Rifki; 30 Grissom Drive, Clifton Park, NY 12065 (US). STILLMAN, Jonathan, Peter; 305 Sweetman Road, Ballston Spa, NY 12020 (US). CORMAN, Jennifer, M.; 2222 Budd Terrace, Schenectady, NY 12309 (US).

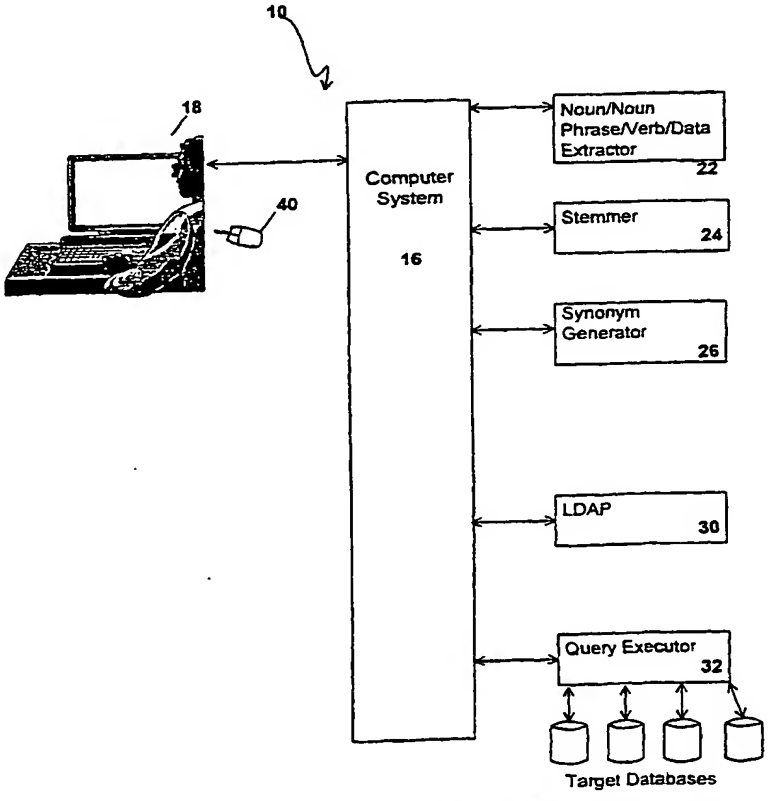
(74) Agent: MARSH, Thomas, R.; Marsh Fischmann & Breyfogle LLP, Suite 411, 3151 South Vaughn Way, Aurora, CO 80014 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian

*[Continued on next page]*

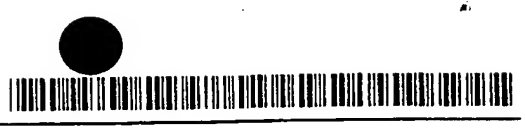
(54) Title: METHOD AND SYSTEM FOR UNIVERSAL QUERYING OF DISTRIBUTED DATABASES




(57) Abstract: A method and system for accessing information or data from distributed databases. The system includes a computer (16) adapted to present the user a query input screen displayable on screen (18), an extractor (22) to extract important nouns or key nouns and noun phrases in a user's query, a stemmer (24) to return the base word in each query term, a synonym generator (26) which identifies synonyms for each query term. The method includes the steps of processing a query to generalize and expand the query to return as many relevant terms to the user, receiving from the user selected terms which the user expects to find in attributes of the distributed databases and searching the directories of the distributed databases using a Lightweight Directory Access Protocol (LDAP) (30). A query executor or mediator (32) retrieves data from the specified databases in accordance with SQL code.

WO 01/41002 A1

BEST AVAILABLE COPY



patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM). European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR). OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

**Published:**

— *With international search report.*

— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

# METHOD AND SYSTEM FOR UNIVERSAL QUERYING OF DISTRIBUTED DATABASES

## FIELD OF THE INVENTION

The present invention generally relates to a system and method for searching target databases, and in particular, to a system and method for universal querying of distributed databases.

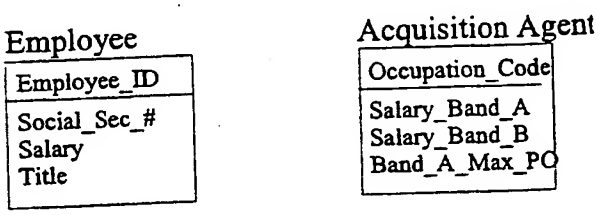
## BACKGROUND OF THE INVENTION

Numerous independently owned collections of data are being created and maintained all over the world. The number of active and legacy data sources almost guarantees that part or all of a query can be answered using one of these countless databases. However, there exist several intermediate steps between posing a query and receiving an answer that make the task of querying others' databases almost impossible for the average user. First, the user must locate a relevant data source. Then he or she must gain access to the source, pose the query using table names and attribute names from that target database, and finally must decide which of the returned data, if any, is relevant to the query. Users' queries must be formatted correctly, either using structured query language (SQL) code, or using formatted blocks of code (i.e., code generated by a back-end process based on user-filled selection boxes and text fields).

While this list of steps is formidable, the process of querying is even more difficult if multiple databases must be consulted to obtain a complete answer. Not only must the above steps be executed, but the data from different sources must be joined; and, if there are discrepancies, the user must decide which source is more reliable. In integrating the data, users must first understand elements of each database's schema so that corresponding fields between databases can be identified. Even once corresponding fields have been located, user must consider both the relative accuracy of the sources and the timeliness of the data contained within the sources. For example, data in a five (5) year old database would obviously be less relevant to data in a current database if a Department of Defense (DoD) member is querying about current troop movements.

There are even more basic problems standing between the user's query and an answering data set. Databases are created with a particular task in mind. The database may be tailored for ease of asking particular types of queries, for ease of storing new data, or for storing groups of attributes as an object. Designing databases for specific purposes allows

data to be stored and retrieved efficiently for that particular task and possibly a few related tasks. However, this makes it nearly impossible to retrieve information for other unrelated tasks. In looking at the task of querying from this perspective, it can be seen that the most fundamental querying problem is that groupings of objects that make sense in one database representation, make it difficult to regroup attributes to form objects meaningful to a query unrelated to the database's specific purpose. For example, consider the database tables below which have been excerpted from a hypothetical company's relational database:




Tables and attributes from a hypothetical company database. The "Employee" table has key Employee_ID and attributes Social_Sec_#, Salary, and Title. The "Acquisition_Agent" table has key Occupation_Code and attributes Salary_Band_A, Salary_Band_B,and Band_A_Max_PO.

A division of this hypothetical company has a database that keeps track of its employees. The database has a table, "Employee," that contains basic information such as name, social security number, salary, and job title. The key in this table is Employee_ID. The database also has individual tables relating to each job title within the company. These tables note the occupation's salary ranges (e.g., Salary_Band_A) and the specific duties at each salary level (e.g., Band_A_Max_PO). For example, for an "Acquisition_Agent," the salary bands are A, B, etc., and the maximum amount that an individual in salary band A may purchase is Band_A_Max_PO. This table's key is Occupation_Code. A reasonable query from another division of the company could be "Return the individuals who can purchase more than 5000 units of product X." Given the above two tables from the database, we can see that the query will be difficult to execute. First, the individual asking the query would have to know that Acquisition_Agent and Buyer were synonymous. Next, a join on salary would need to be executed, but there is no common key. Finally, math would have to be performed to translate between the maximum purchase order allowed (Band_A_Max_PO)



BNSDOCID: <WO 0141002A1_I_>

and the number of units of X a specific buyer could purchase. This seemingly simple query requires a great deal of database-specific knowledge.

From the above discussion it is clear that there can be a number of issues encountered in trying to retrieve data from an unfamiliar source or sources. There is the initial task of locating relevant data sources. Even once this has been accomplished, the problem of answering the query becomes no easier. Issues range from the banal, but nontrivial, task of gaining access privileges, to the more theoretical and complex tasks of regrouping of attributes to form real-world entities (i.e., the attributes within a table must be understood as representations of actual physical objects). Several potential obstacles are discussed below.

The first potential obstacle concerns gaining access to the relevant data source. This involves being allowed to read the database schema and the data contained within the database. Additionally, it may require the ability to store intermediate tables. When a large, multi-step query with several joins or cross products is carried out, the intermediate tables generated need to be temporarily stored. If systems accessing the database are remote, it is clearly impractical to transmit these larger data sets to the querying machine. Thus, some local write space may be desired.

A second potential obstacle concerns the fact that each database in the system may have been designed for efficiency for a system-specific task. Databases are created to fit within larger systems. These systems have certain storage and retrieval requirements, as well as baseline assumptions about data format. No matter how general a database schema is developed, the schema must operate within the system and data requirements. This necessarily means there are queries the system will have difficulty answering.

A third potential problem is that poorly labeled tables and attributes can make it impossible to determine the real-world object being represented. Examples of table names extracted from actual DoD data sources include: $UD01, VNNZ, SYFA, and WUC1. Examples of attribute names extracted from the same DoD source include: SC, TCN, FROM_PPC1, and PRIME. Without the aid of documentation or the original database designers, it is impossible to know what physical objects are represented by these tables. Thus, data corresponding to a user's query is forever lost because a user or an automated system will be unable to identify all relevant data.



BNSDOCID: <WO___0141002A1_I_>

The fourth potential problem in trying answer a query is that documentation is typically scarce and may not be any less cryptic than the database objects themselves. Additionally, original database designers may have forgotten what the objects represent, or they may have moved onto other sites. Users are left to map between database schema and real-world objects to the best of their ability.

If the average user is able to overcome these obstacles and retrieve data from several data sources, he must then combine the responses into a coherent solution set. This compilation may involve conflict resolution among data rows. In some situations, it may be acceptable to return both data items and allow the user to decide which data item is more reliable. Consider however a fictitious military example. Two different databases return different locations for the same enemy tank. One location is very close to a US Army base, and the other set of coordinates places the tank much farther away. How should the Army General querying the system react? Should he or she assume the tank is close and ready the troops, and thus risk looking as if the base is preparing for military action? Or should the General not mobilize troops and risk being unprepared for an enemy attack? If the General does not know which data is more accurate in this case, then it is very difficult to determine which results are correct and what action to take.

In addition to the issues discussed in the previous section, attempting to locate relevant databases and achieve accurate query responses in a military environment can be even more difficult. For example, not only does the user need to gain access to a database, but he or she must typically have the appropriate clearance level to see every row and column of the data returned. An even bigger obstacle to overcome is the fact that terminology across branches of the military is not always consistent. First, the same term may have different meanings in different divisions of the military (e.g., rank has different meanings across government military components). Second, the same object (e.g., a 20-foot antenna or a type of ammunition) can have different names in different branches. The first issue leads to a problem in query interpretation while the second creates a problem in retrieving data across databases.



BNSDOCID: <WO 0141002A1 I >

## SUMMARY OF THE INVENTION

Accordingly, it is an object of the present invention to provide a system and methodology for querying distributed databases.

It is another object of the present invention to provide a query system and method adapted to process unstructured queries.

It is a further object of the present invention to provide a querying system and method which allows a user to retrieve data from a database as soon as such database is introduced into the system without causing the system to be halted or rebooted.

It is still another object of the present invention to provide a querying system and method which aids in the generation of mediators.

It is still another object of the present invention to provide a querying system and methodology which does not utilize a shared representation.

Generally, the system and method of the present invention accomplishes one or more of the above-noted objects of the present invention by providing an architecture which allows users to enter unstructured queries, expands and generalizes such queries, and matches the queries to actual target database tables. The method of the present invention generally includes the steps of processing a query (e.g., an unstructured query) to generalize and/or expand the query to return as many relevant words or terms as possible to the user, receiving from the user selected words or terms which the user expects to find in attributes of the distributed databases, and searching a database structure (e.g., an annotated database) having directories extracted from target distributed databases, the directories including table names, attribute names, sample data, and/or, if available, data dictionary information. Of importance, the step of searching the database structure includes utilizing a Lightweight Directory Access Protocol (LDAP), which allows quick access to information directories. Since LDAP directories are designed for reading data rather than updating or adding new data to the directories, the retrieval speed of information contained within the directories (e.g., table names and table attributes) is very fast.

In one aspect of the method of the present invention, the step of processing a query includes the step of receiving at least a first query from a user or client, the first query including at least a first term and the steps of identifying key terms and generalizing and/or expanding the first query to enhance the likelihood of retrieval of relevant data to the user. In this regard, the step of processing at least the first query may include the step of

identifying or extracting key words or terms from the first query, such as the first term, since such key words may correspond to an attribute or table name in the target distributed databases. In one embodiment, the step of extracting key words includes the step of extracting at least a first noun and/or a first noun phrase from the first query. In another embodiment, the step of extracting key words comprises the step of extracting at least a first verb from the first query. In yet another embodiment, the step of extracting key words comprises the step of extracting at least a first data item (e.g., part number) in the first query. In order to further generalize the first query in order to enhance the chances of capturing relevant information from target databases, the step of processing at least the first query comprises the step of stemming at least a first term in the first query, such that at least a first root word corresponding to the first term may be utilized in the final search. The processing step may also include the step of generating at least a first synonym of at least the first term of the first unstructured query to expand the scope of the search. The step of processing at least the first query may be facilitated by presenting to the user an initial user query screen, whereby the user is afforded an opportunity to perform various options, including perform stemming, include synonyms, include acronyms, and/or perform wild card substitutions.

Once such nouns, noun phrases, verbs, numbers, synonyms, acronyms, and/or related terms are retrieved and/or generated, the processing step may include the step of presenting such terms to the user in an expanded or refined user query screen format. Such relevant words (e.g., nouns, noun phrases, verbs, numbers, and/or synonyms) may be presented to the user or client to afford the user the opportunity to select the returned relevant terms (e.g., gathered nouns, noun phrases, synonyms, acronyms, data items and/or related items) which the user believes useful in searching the target distributed databases. As a result, the user is able to select or collect terms for which the database schemae will be searched.

In order to facilitate subsequent searches by a user, the step of processing the first unstructured query from the user may further include the step of ranking selected relevant terms (e.g., synonyms or other related terms). In this regard, if a term is selected, the rank for such term is increased and, conversely, if a term is not selected, the rank for such term is decreased. Additionally, the methodology of the present invention is adapted to learn from the structure of the users' queries. In this regard, if query terms frequently occur together,

when a user submits only one of these terms, information regarding both terms may be returned to the user to save the user time. Such learning capability may be included within the LDAP. Conversely, if certain synonyms are not frequently selected, such synonyms will not be returned to users in the future.

As noted hereinabove, the method generally includes the step of searching a database structure, such as an LDAP directory which may include attributes, table names, sample data and/or data dictionary information in the target distributed databases, for attributes and/or table names that match the terms selected by the user (e.g., augmented query terms) and presenting such information to the user. Such attributes and/or table names may be retrieved, along with the remaining attributes for tables that had matching attribute names. A first tree may be constructed, whereby query term folders are populated with database folders containing the tables that match the augmented query terms and such tree is presented or returned to the user. Such folders are labeled with the query term or terms that correspond to the matching tables contained in them. The methodology of the present invention may further include the step of processing a final query from the user. In one embodiment, the step of creating a final query comprises creating a pictorial query for the user, whereby the user is allowed to add constraints and/or joins to produce a final query. The step of processing the final query further includes the step of automatically generating SQL code corresponding to the final query and utilizing a mediator to forward the query to appropriate databases, receive data from each of the appropriate databases, and returning such data to a servlet, where such data is formatted and presented to the user.

In another aspect, the present invention relates to a system for processing at least a first query to retrieve data relevant to the first query from at least a first of a plurality of distributed or target databases. Generally, the system of the present invention includes a computer system for at least receiving a first query from a first user, an extractor for identifying key words, such as nouns, noun phrases, verbs or numbers in the first query, a database structure, such as an LDAP directory, including at least one of a plurality of table names and attributes relating to tables within the distributed databases, the directory being searchable to provide the user, via the computer system, with at least a first database table name and attributes associated with retrieved tables corresponding to the retrieved table names, and a code generator for generating SQL code based upon retrieved table names and/or attributes selected by the user. In order to enhance the search, the system may



BNSDOCID: <WO 0141002A1 I >

further include a query generalizer for processing the first query to provide or return to the user via the computer system terms related to at least a first term of the first query to enable the user to select terms the user expects to find in distributed database tables. For purposes of facilitating searching, the system may further include a learning program, whereby information about which synonyms a particular type of user will need and which terms often appear together in these queries is stored and/or ranked. In this regard, after the user enters a query and chooses relevant synonyms, the rank of terms is updated. If terms are selected, the rank of such terms is increased and conversely, if terms are not selected, the rank of such terms is decreased. Further, the system is adapted to learn from the structure of the users' queries. In this regard, if terms frequently occur together, then when the user asks only about one of these terms, the system will return information about those terms to the user. The system may further include a central mediator for receiving the query and SQL code, via any computer system, the central mediator in communication with the target distributed databases. Such central mediator may be adapted to return the retrieved data from the appropriate distributed databases to the computer system, which is capable of formatting and presenting such data to the user via, for example, a display screen.

## BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a diagrammatic illustration showing one embodiment of the system of the present invention;

Fig. 2 is a diagrammatic illustration showing the architecture of the system illustrated in Fig. 1;

Fig. 3 illustrates a view of displayed textual and graphic information related to entering a query via the screen;

Fig. 4 illustrates the structure of a sample LDAP directory;

Fig. 5 illustrates a view of displayed textual and graphic information related to expanding a user's query via the screen;

Fig. 6 illustrates a view of displayed textual and graphic information related to finalizing a user's query via the screen;

Fig. 7 illustrates a sample SQL query;

Figs. 8A-8B present a flow chart of one embodiment of the method of the present invention; and



BNSDOCID: <WO 0141002A1 I >

Figs. 9A-9D present a flow chart of another embodiment of the method of the present invention.



BNSDOCID: <WO 0141002A1 I >

## DETAILED DESCRIPTION

Generally, the method and system of the present invention allows users or clients to enter an unstructured query that the system expands and generalizes and then matches to actual database tables. Users may interact with the system in three different ways. First, the users may enter an unstructured query, which may be a list of important terms as in a typical search query or, alternatively, the query may be a natural language question or sentence. Second, users may select the nouns, noun phrases, synonyms, and/or related terms that the user expects to appear in the table names and/or attribute or field names of the target databases. In this regard, after the unstructured query is received, nouns and/or noun phrases in the query may be identified, and the query may be generalized and/or expanded to return to the user as many relevant words as possible. From these returned words, the user may select the terms he or she expects to find in the database attributes and/or table names in the system. And third, the user may form Structured Query Language code ("SQL") by clicking on tables and attributes presented to the user. In this regard, the database matches that the system believes correspond to the query are presented to the user and given the tables, the user may form a pictorial query from which the SQL code is automatically generated. Thereafter, the data itself may be displayed to the user.

One embodiment of the system 10 of the present invention is illustrated in Fig. 1. The architecture associated with the system of the present invention is illustrated in Fig. 2. As noted hereinabove, the method first involves an interface with the user to allow the user to specify an initial query. The first step in the querying process begins with the user entering an initial query and ends with the systems returning nouns, noun phrases, verbs, data synonyms, and/or related terms. The purpose of this phase of the querying process is to identify key words of the query and to expand and/or generalize the initial query so that later in the process, as many correct tables as possible may be returned to the user. In particular, the system 10 of the present invention includes a computer system 16 (e.g., a servlet), which generally functions to receive and send data to the appropriate destination. In this embodiment, the computer system 16 is adapted to present the user with a query input screen 50, illustrated in Fig. 3, which is displayable to the user via a display screen 18, illustrated in Fig. 1. Such user query input screen 50 allows the user to enter an unstructured query in the query field 52, illustrated in Fig. 3. Via this user query input



BNSDOCID: <WO 0141002A1 | >

screen 50, the user may select which particular generalization and/or expansion functions should be utilized in expanding the query, such as perform stemming, include synonyms, include acronyms, perform wild card substitution and enable user assisted learning (to be described in more detail hereinbelow). In this regard, the system 10 also includes a noun/noun phrase extractor 22, a stemmer 24 and a synonym generator 26. The noun/noun phrase extractor 22, which is commercially available from various vendors, is adapted to identify important or key nouns and noun phrases in a user's query. In general, because of the way in which queries are phrased, the most important items in a user query are nouns, noun phrases, conditionals (e.g., greater than) and numbers. Thus the noun/noun phrase/verb/data extractor 22 searches for only nouns, noun phrases, verbs and data and returns to the user, via the computer system 16, a list of queryTerms (i.e., identified or extracted nouns/noun phrases/verbs/data). The stemmer 24 returns the base word for each queryTerm in the returned list. For example, the queryTerm "guns" would be turned into "gun" and "buyers" would be turned into "buy."

The system 10 is also adapted to expand each term by finding synonyms. In this regard, the synonym generator 26, which is commercially available as Princeton's WordNet Lexicon, identifies synonyms for each queryTerm, and those synonyms whose rank is greater than the system threshold are collected and returned to the user. Initially, all terms are ranked at zero (0). As terms are returned and selected by the user, the rank of such selected terms increases. If returned terms are not selected by the user, the rank of such non-selected terms decreases. When the rank of such non-selected terms falls below the user-acceptability level, these terms are no longer returned to the user. In the event user assisted learning is enabled or desired by the user, related terms may be retrieved from the LDAP 30, such related terms being words that frequently occur in queries with a queryTerm. In this embodiment, the LDAP directory's learning branches, an example of which is illustrated in Fig. 4, may contain the most frequently co-occurring terms for every queryTerm. More specifically, as users query the system 10, the choices they make are used to learn what types of information should be returned to them in future queries. Initially, the system 10 knows nothing. All users have a blank slate, and all query responses are equally "good" answers. After users enter queries and choose relevant synonyms, the rank of terms is updated. If terms are selected, their rank is increased. Conversely, if terms are not selected, their rank is decreased. Only terms whose rank is greater than the system

threshold are returned. Additionally, the system 10 is adapted to learn from the structure of the users' queries. If terms frequently occur together (e.g., gun and ammunition), then when the user asks only about one of these terms, the system 10 will return information about both. This saves the user time from having to add terms and resubmit the query since the information may be already provided. These co-occurrence terms also suggest related information to the user and thus expand users' queries. All this information is kept for each type of user so that different meanings across different entities (e.g., Department of Defense components) may be remembered. Once the nouns, noun phrases, verbs, data, synonyms and/or related terms are retrieved, they are formatted (e.g., by the computer system or servlet 16) into a Java Swing class component that allows data to be displayed in a tree format (called a DefaultMutable Tree) and sent to the expanded user query screen, illustrated in Fig. 5, which is displayable to the user via display device 18.

Fig. 5 illustrates the expanded user query screen 60 which contains the nouns, noun phrases, synonyms and/or related terms which have been retrieved and displayed in a tree format in a left-hand area 62 of the screen. In this expanded user query screen 60, users can open folders, revealing synonyms and related terms associated with different queryTerms. The right-hand area 64 of the screen serves as an area to collect terms for which the system's database schemae will be searched. Users may add individual terms or entire folders in the left-hand area 62 to the right-hand side of the screen. The items in that area 64 will be searched for in the attribute names of the target databases . Additionally, users may enter new queries in the Query box 66 and may either choose to expand and generalize these terms and return to the screen to continue adding terms to the right-hand side 64 (Refine), or may choose to Submit the query. If Refine is chosen, the steps discussed hereinabove relating to generalization and expansion of the query are followed, and the new tree of terms, synonyms, and related terms is sent to the user. However, if Submit is selected, the terms in the Query box 66 are expanded and generalized as above, and the intermediate tree is not displayed to the user. Instead, all nouns and noun phrases, verbs, data, synonyms and/or related terms are appended to the list of terms retrieved on the right-hand area 64 of the expanded user query screen 60, illustrated in Fig. 5. Once the user submits these queryTerms, wild cards may be added to the beginning and end of each term, and spaces are replaced with wild cards (if the queryTerm is a noun phrase). Thereafter, the LDAP structure 30, illustrated in Fig. 1, may be searched, and all attributes in the target



BNSDOCID: <WO 0141002A1 >

databases that match the augmented queryTerm are returned. The system retrieves the rest of the attributes for the tables that had matching attribute names, and a tree is constructed where queryTerm folders are populated with database folders containing the tables that match. This tree is then returned to the user via the computer system or servlet 16 as the query generating screen 70, illustrated in Fig. 6.

Of importance, the LDAP protocol allows quick access to the information in its directories, and provides capabilities allowing greater expression searches. Because LDAP directories are designed for reading data rather than updating or adding new data to the directories, the retrieval speed is fast. A sample LDAP structure 80 is illustrated in Fig. 4. The left half 82 of the LDAP hierarchy holds data used in the learning algorithm (e.g., for retrieval of related terms and/or synonyms). Data in these branches is associated with specific types of users (e.g., general users, Navy officers, etc.). The right half 84 of the tree contains details about databases contained in the system. For each database, all tables, attributes, attribute data types, and whether or not the attribute is a key is stored. The speed of the LDAP search is coupled with the regular expression querying capability is the primary driver for storing both database structure and learning information in the LDAP. As note hereinabove, the directory tree has two halves, a learning half and a database half. In the learning half 82, information is stored about which synonyms a particular type of user will need and which terms often occur together in these queries. In the database half 84 of the tree, the structures of all the databases in the system are stored. In this embodiment, there are four databases in the system: SUPPLY, Rainbow, KVDBA, and KVAWUB. Such databases may be extracted from structure databases (e.g., Oracle), or any other similar database.

The third general step in the querying process is directed to creating a final query. More specifically, this phase of the querying process allows users to create a pictorial query and add constraints, view the automatically generated SQL code, submit the query, and browse the resulting data set. As noted hereinabove, the system 10 illustrated in Fig. 1 is adapted to construct a tree whereby queryTerm folders are populated with database folders containing the tables that match. Once the suggested tables are returned to the user via the query generating screen 70, the user may select tables from the left-hand area 72 of the query generating screen 70, adding them to the right-hand work area 74. As the tables 76a, 76b are added, keys or attributes that match between tables are automatically connected

with lines (e.g., joins 78), both of which may be deleted by the user, if desired. The system 10 is adapted to allow users to create further joins by clicking and dragging a mouse 40, illustrated in Fig. 1, between attributes in different tables. Users may also add constraints on a specific attribute by clicking the attribute so that it appears in the bottom portion 79 of the screen, then filling in the rest of the constraint. Once the query is built by the servlet 16, the user may view the SQL code in another screen presentable to the user (an example of which is illustrated in Fig. 7, and press or select the submit option. The servlet 16 then passes the query to a query executor 32 (e.g., mediator) which is commercially available from a variety of vendors, the mediator being adapted to retrieve data from the specified databases in accordance with the SQL code. The data retrieved are then returned to the servlet 16, where the data is formatted and presented to the user via the display device 18. In this embodiment, the query executor or mediator, in accordance with the SQL code, is directed to retrieve only data from particular databases, tables and attributes, in view of the joins and constraints, if any. No mappings between databases are created.

In another aspect, the present invention is directed to a method for querying distributed databases. Generally, and referring to Figs. 8A-8B, in this embodiment, the method of present invention includes a step 112 of receiving from a user a query (e.g., structured query or unstructured query). Thereafter, the method includes a step 116 of identifying/extracting nouns, noun phrases, verbs and/or data from the query. Thereafter, the methodology includes a step 120 of sending to the user the enhanced query, at which point the user may select terms which will be searched for any attribute names of the target databases. Users are also afforded the opportunity to enter new queries. In this regard, in the event a new query is received from the user, the new query may be received and then processed as described herein-above. Otherwise, the method includes the step 124 of receiving the selected terms of the enhanced query from the user. Thereafter, the method includes the step 128 of searching the database structure (e.g., LDAP structure) to retrieve all attributes in the target databases that match the terms of the enhanced query selected by the user. The method then includes the step of retrieving the rest of the attributes for the tables that had matching attribute names and the step 132 of sending to the user matching tables in the distributed databases. At this point, the method of the present invention allows the user to create a pictorial query and to add constraints. In this regard, the method includes the step 136 of receiving selected tables from the user corresponding to the table



RNSDOCID: <WO 0141002A1 | >

from which the user wishes data to be retrieved from the system databases. Upon receipt of such selected tables from the user, the method includes the step 140 of generating an SQL code and the step 144 of retrieving data from the appropriate target databases and the step 148 of sending the retrieved data to the user.

Another embodiment of the methodology of the present invention is illustrated in Figs. 9A-9D. Initially, the methodology includes the step 210 of presenting the user with a query input screen, which is illustrated in Fig. 3. As noted hereinabove, the query input screen presents the user with an area to enter the query, and gives the user several options, such as whether to perform stemming, include synonyms, include acronyms, perform wild card substitutions, and enable user assisted learning. In this regard, the method further includes the step 214 of receiving the initial query from the user and the step 218 of extracting nouns and/or noun phrases from the initial query, such extracted nouns/noun phrases identified as "queryTerms". In the event the user has requested stemming to be performed, at least a first noun in the initial query or a first queryTerm is stemmed (step 222), and in the event the user has requested synonyms be included, the method may further include the step 226 of generating at least a first synonym relating to the first noun or first queryTerm. In the event the user assisted learning has been enabled or requested by the user, the method includes the step 228 of retrieving related terms from the LDAP, a related term being a word that frequently occurs in queries with a queryTerm.

Once the nouns, noun phrases, verbs, data, synonyms and/or related terms are retrieved, the method further includes the step 230 of presenting such terms to the user in an expanded user query screen, illustrated in Fig. 5. As noted hereinabove, the expanded user query screen allows the user to specify which synonyms and/or related terms correspond to the query. In this regard, the method further includes the step 234 of receiving stem terms, related terms and/or synonyms selected by the user. The method then includes a step 138 of ranking the selected and non-selected terms or words and the step 142 of storing the rank of such terms and the related terms for future use (as described hereinabove). Utilizing the queryTerms selected by the user, the method includes the step 246 of searching the LDAP for matching attributes and the step 250 of retrieving the matching attributes and associated table names.

The method then includes the step 254 of presenting to the user a query generating screen, illustrated in Fig. 6. This phase of the querying process allows users to create a

pictorial query, add constraints, view the automatically generated SQL code, submit the query and browse the resulting data set. As noted hereinabove, the query generating screen illustrated in Fig. 6 presents to the user the suggested returned tables, from which the user may select by adding selected tables to the right-hand work area from the left-hand side of the screen. It should be noted that in the event the user is not satisfied with the presented/retrieved tables, the user may elect to refine the query, whereby the user will be presented with the expanded query screen, illustrated in Fig. 5, or, alternatively, the user may submit a new query by going back to the user query input screen. In the event the user wishes to continue, the method further includes the step 256 of joining matching attributes between selected retrieved tables. In the event the user does not wish to proceed with the query utilizing one or more of the automatically joined matching attributes, the method includes the step 260 of deleting appropriate joins, as selected by the user. The user may also add joins manually by clicking and dragging the mouse between attributes in different tables. In this regard, the method may further include the step 264 of joining selected attributes, as selected by the user. The method of the present invention further allows constraints to be added on specific attributes by clicking the attribute so that it appears in the bottom portion of the query generating screen, and allowing the user to fill in the constraint. In this regard, the method includes the step 268 of constraining selected attributes in accordance with the user's request. Thereafter, the method includes the step 272 of generating an SQL query based upon the selected tables, constraints and joins. Such SQL code may be presented to the user at step 276. Once the query is built, the user may view the SQL code and submit the query to the servlet, which passes the query to the mediator which is adapted to retrieving data from the specified databases in accordance with the SQL query. In this regard, the method includes the step 280 of receiving the SQL query, the step 284 of retrieving data from the appropriate target databases in accordance with the SQL query, the step 288 of formatting retrieved data and finally, the step 292 of presenting the formatted retrieved data to the user.

As a result, the present invention is particularly useful in aiding users in accessing data from distributed, structured databases, whereby users need not know the structure or even the existence of the databases needed to complete their queries. When querying the system, users need not know of the existence of relevant data sources currently available in the system, need not understand the schemae of the databases, need not know SQL, and are

not limited to formatting queries using drop-down menus. Rather, users may enter an unstructured query, select from synonyms and related terms automatically generated by the system to expand the user's initial query, and then generate a pictorial query using database tables the querying system suggests as relevant. After forming and submitting this query, users are presented with the corresponding data from actual databases.

The foregoing description of the present invention has been presented for purposes of illustration and description. Furthermore, the description is not intended to limit the invention to the form disclosed herein. Consequently, variations and modifications commensurate with the above teachings, and the skill or knowledge of the relevant art, are within the scope of the present invention. The embodiments described hereinabove are further intended to explain best modes known for practicing the invention and to enable others skilled in the art to utilize the invention in such, or other, embodiments and with various modifications required by the particular applications or uses of the present invention. It is intended that the appended claims be construed to include alternative embodiments to the extent permitted by the prior art.

CLAIMS

What is claimed is:

1. A method of obtaining information from a plurality of distributed databases, comprising the steps of:

receiving a first query from a first user;

processing the first query to identify a plurality of key terms in the first query, the plurality of key terms comprising at least one of a first noun and a first noun phrase;

displaying to the first user an expanded query including at least a plurality of returned key terms;

receiving from the first user a plurality of select key terms, the plurality of select key terms being selected from the plurality of returned key terms by the first user;

processing the expanded query to retrieve a plurality of attributes and a plurality of table names, wherein the plurality of attributes corresponds to the plurality of table names, wherein the plurality of table names correspond to a plurality of tables in the plurality of distributed databases;

displaying to the first user the plurality of table names and the plurality of attributes;

receiving from the first user a final query, the final query including a plurality of select tables, the plurality of select tables being selected from the plurality of table names by the first user;

processing the final query to generate a first SQL query corresponding to the final query; and

returning to the first user a first data result set based on the first SQL query.

2. A method as claimed in Claim 1, wherein the first query is an unstructured query.

3. A method as claimed in Claim 1, wherein said step of processing the first query comprises the step of stemming a first of the plurality of key terms.

4. A method as claimed in Claim 1, further comprising the step of:

identifying at least one of a first verb and a first data item within the first query.

5. A method as claimed in Claim 1, wherein said step of processing the first query comprises the step of identifying at least a first synonym of a first of the plurality of key terms.

6. A method as claimed in Claim 1, further comprising the step of:

ranking at least each the plurality of select key terms to produce a rank of terms; and

storing the rank of terms for the first user.

7. A method for obtaining data from a plurality of distributed databases, said method comprising the steps of:

receiving a first query from a first user; and

processing the first query to search a plurality of directories corresponding to the plurality of distributed databases to retrieve a plurality of database structures, wherein the plurality of database structures correspond to the first query.

8. A method as claimed in Claim 7, wherein the plurality of database structures comprises a plurality of table names and attributes.

9. A method as claimed in Claim 7, wherein said processing step comprises the steps of:

extracting from the first query at least one of a first noun, a first noun phrase, a first verb and a first number; and

searching the plurality of directories for at least one of the first noun, the first noun phrase, the first verb and the first number.

10. A method as claimed in Claim 7, wherein said processing step comprises the steps of:

stemming at least a first term relating to the first query to produce a first stemmed term; and

searching the plurality of directories for the first stemmed term.

11. A method as claimed in Claim 7, wherein said processing step comprises the steps of:

retrieving for a first synonym of a first term relating to the first query; and

searching the plurality of directories for at least one of the first term and the first synonym.

12. A method as claimed in Claim 7, wherein said processing step comprises the steps of:

retrieving at least a first related term corresponding to a first term, the first term being related to the first query;

receiving from the first user at least a first selected term corresponding to at least one of the first relevant term and the first term of the first query; and

searching the plurality of directories for at least the first selected term, wherein a first of the plurality of directories includes the first selected term.



BNSDOCID: <WO 0141002A1 >

13. A method as claimed in Claim 12, further comprising the step of:

generating a first SQL query from at least the first selected term and the first of the plurality of directories;

retrieving from a first of the plurality of distributed databases corresponding to the first of the plurality of directories first data, wherein the first data is displayable to the first user.



BNSDOCID: <WO 0141002A1 I >

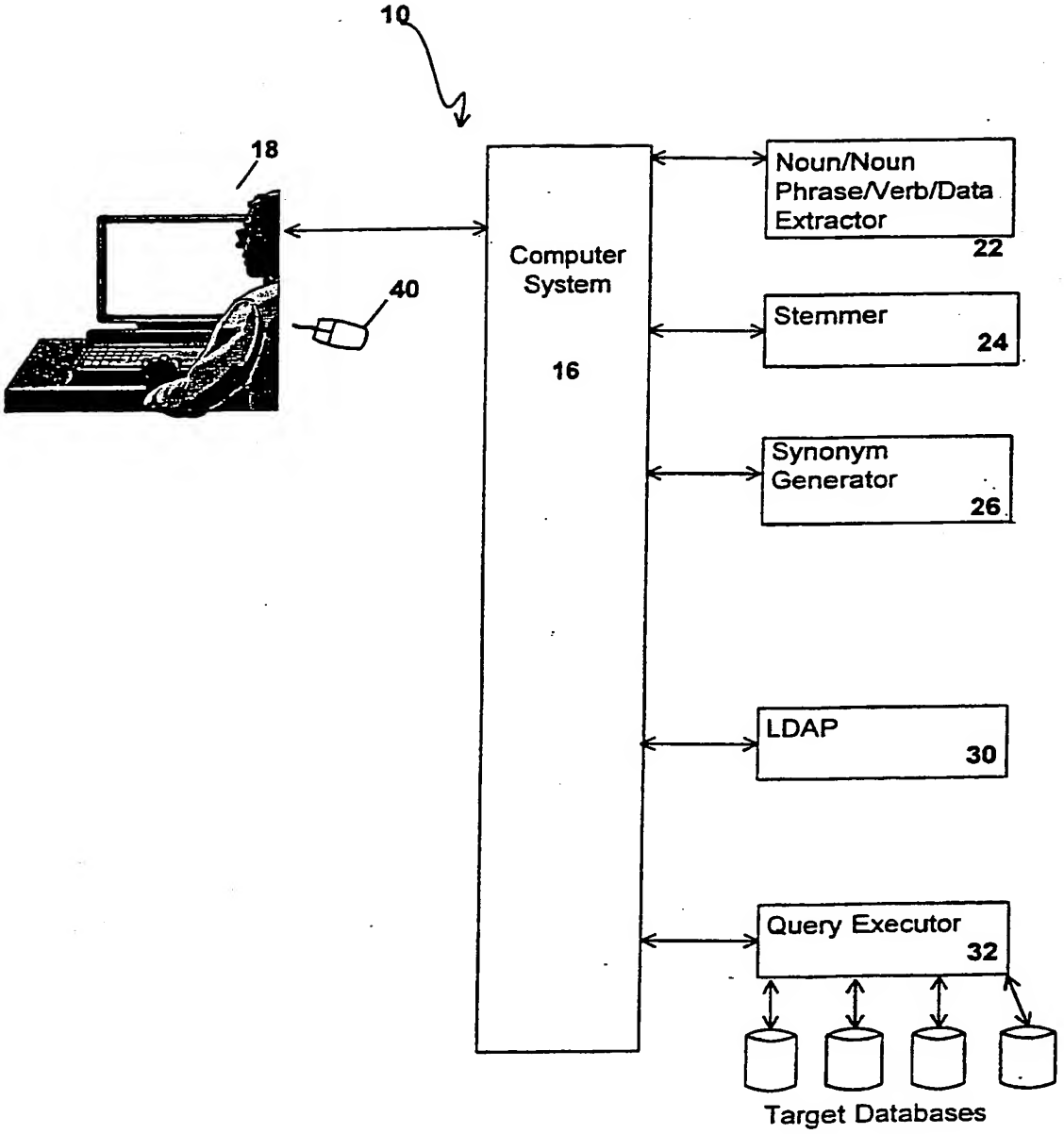

10
18
40
Computer System
16
Noun/Noun Phrase/Verb/Data Extractor
22
Stemmer
24
Synonym Generator
26
LDAP
30
Query Executor
32
Target Databases

FIG. 1

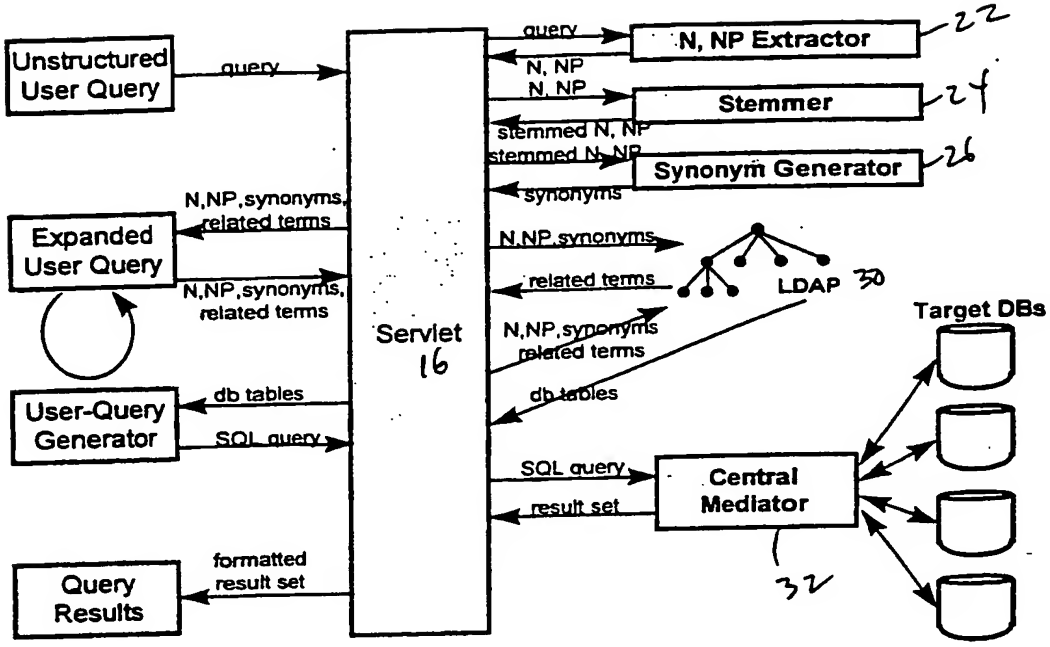

Unstructured User Query
query
Expanded User Query
N,NP,synonyms, related terms
N,NP,synonyms, related terms
User-Query Generator
db tables
SQL query
Query Results
formatted result set
Servlet 16
query
N, NP Extractor
22
N, NP
N, NP
Stemmer
24
stemmed N, NP
stemmed N, NP
Synonym Generator
26
synonyms
N,NP,synonyms
related terms
LDAP
30
N,NP,synonyms related terms
db tables
SQL query
result set
Central Mediator
32
Target DBs

FIG-2

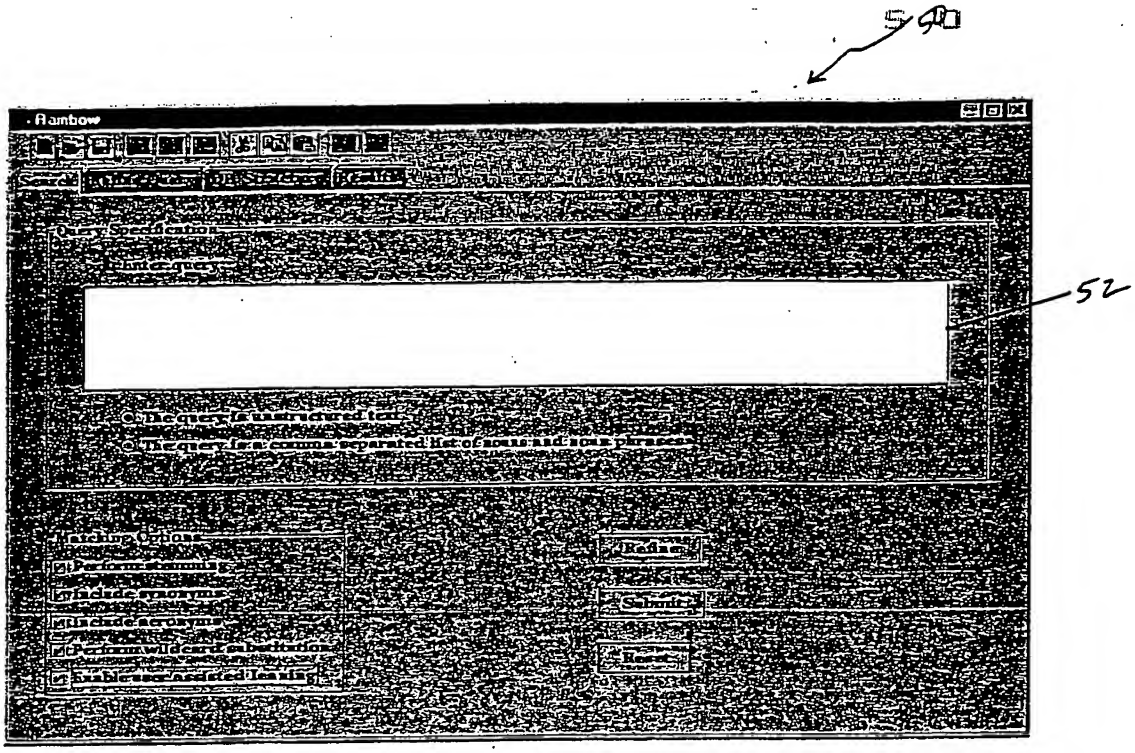



FIG. 3

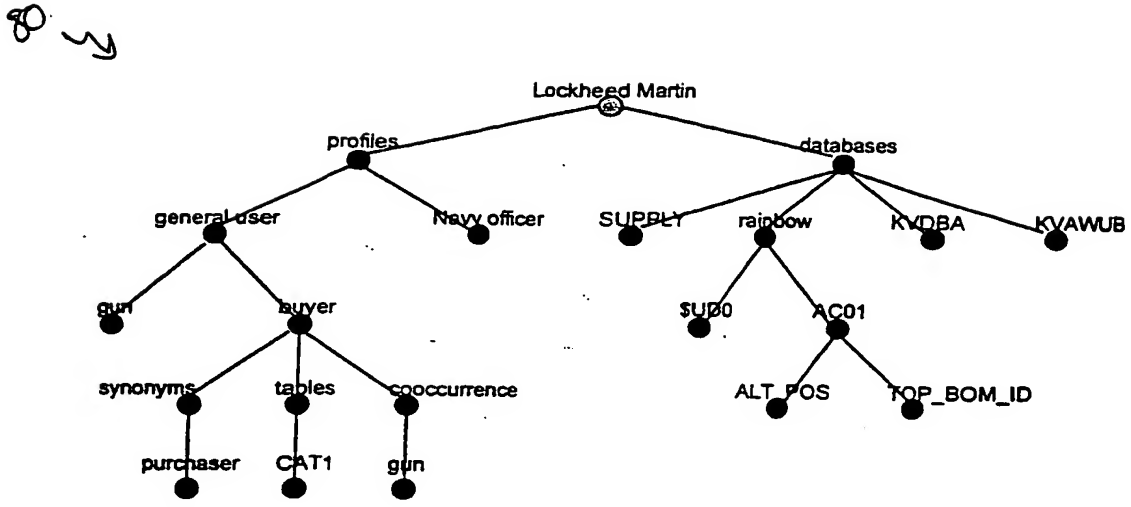

80
Lockheed Martin
profiles
databases
generalUser
Navy officer
SUPPLY
rainbow
KVDBA
KVAWUB
gun
buyer
SUB0
AC01
synonyms
tables
cooccurrence
ALT_POS
TOP_BOM_ID
purchaser
CAT1
gun
82
84

FIG-4

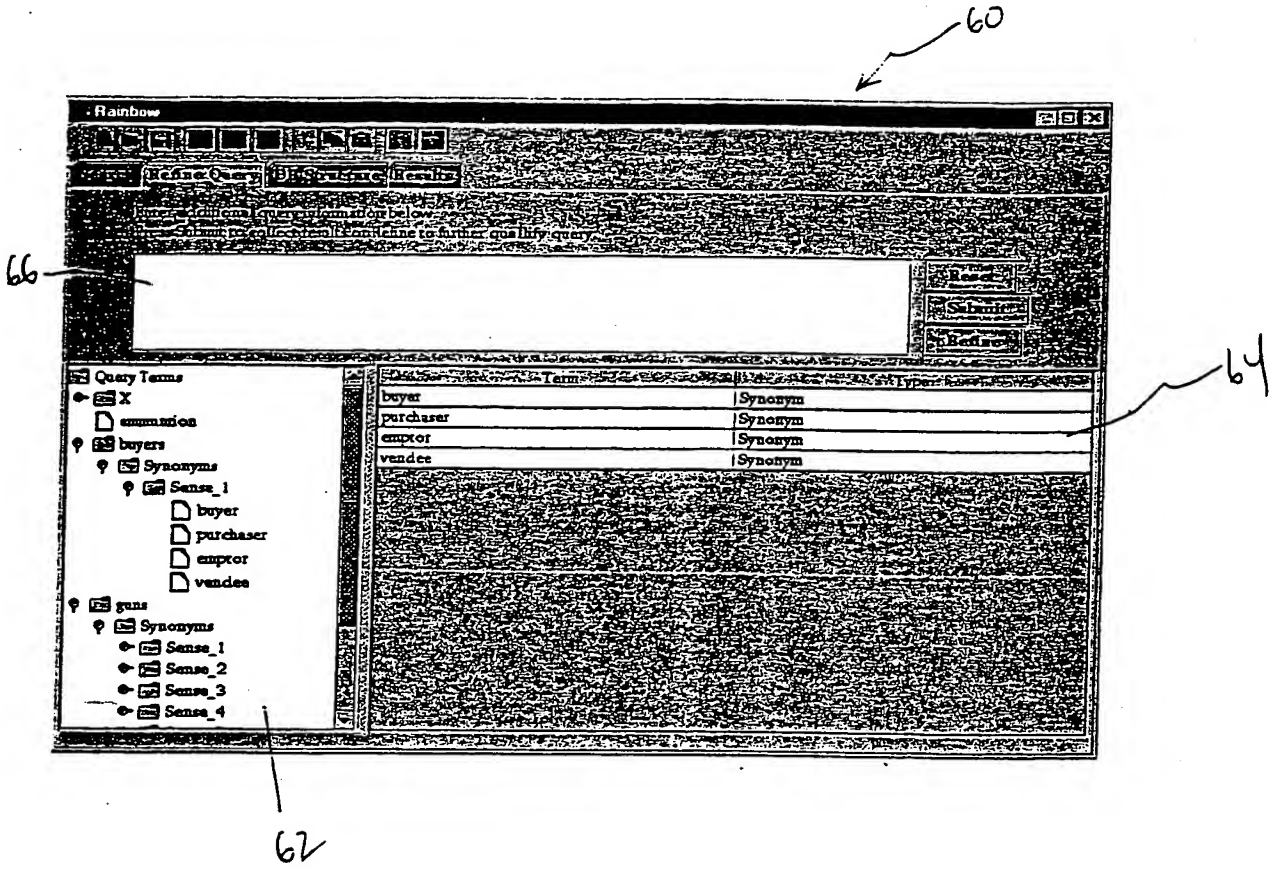

60
Rainbow
66
Reset
Submit
Refine
64
Query Terms
X
buyers
Synonyms
Sense_1
buyer
purchaser
emptor
vendee
guns
Synonyms
Sense_1
Sense_2
Sense_3
Sense_4
Term
Type
buyer
Synonym
purchaser
Synonym
emptor
Synonym
vendee
Synonym
62

FIG-5

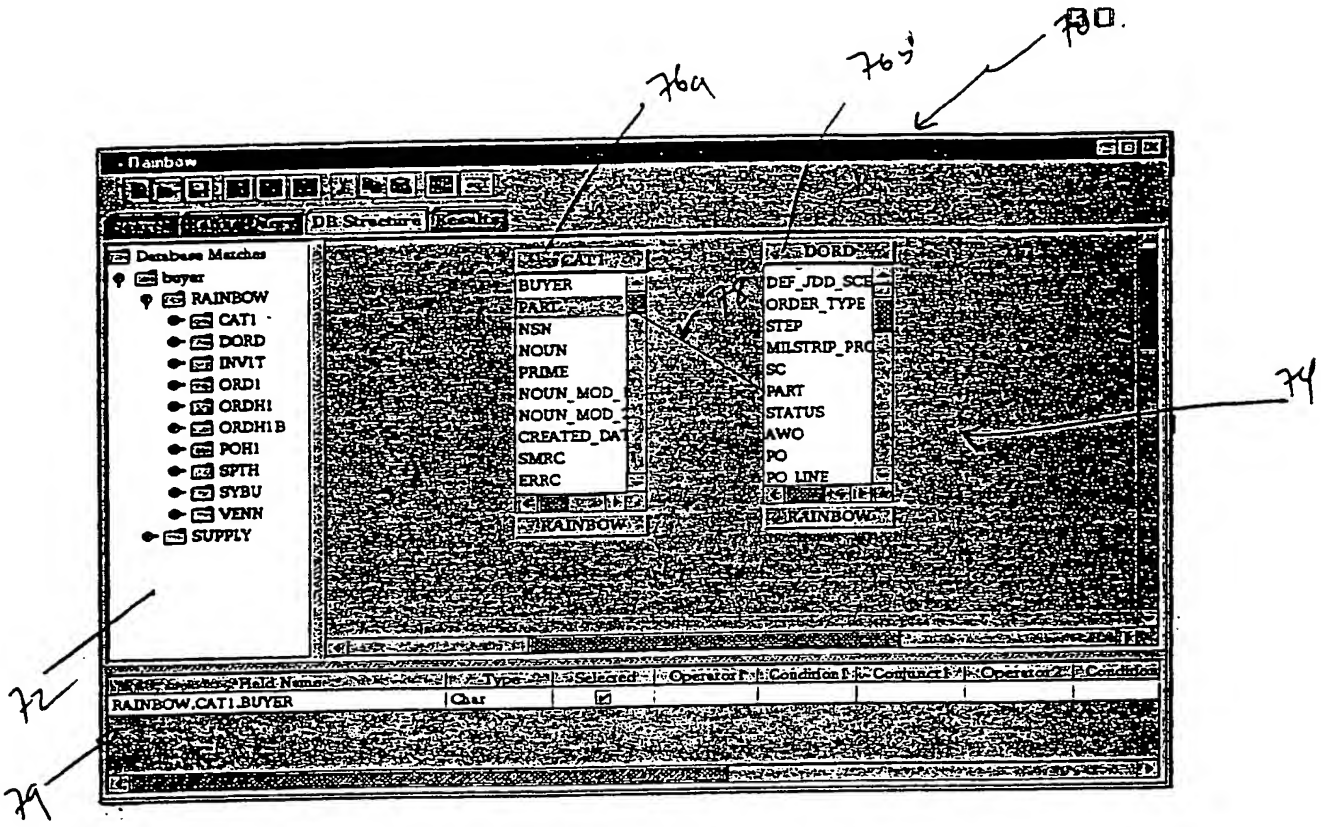

76a
76b
70
74
72
79
Rainbow
DB Structure
Database Meches
buyer
RAINBOW
CAT1
DORD
INVLT
ORD1
ORDH1
ORDH1B
POH1
SPTH
SYBU
VENN
SUPPLY
BUYER
PART
NSN
NOUN
PRIME
NOUN_MOD_
NOUN_MOD_
CREATED_DAT
SMRC
ERRC
DEF_JDD_SCE
ORDER_TYPE
STEP
MILSTRIP_PRC
SC
PART
STATUS
AWO
PO
PO_LINE
RAINBOW
Field Name
Type
Selected
Operator 1
Condition 1
Conjunct 1
Operator 2
Condition 2
RAINBOW.CAT1.BUYER
Char

FIG. 6

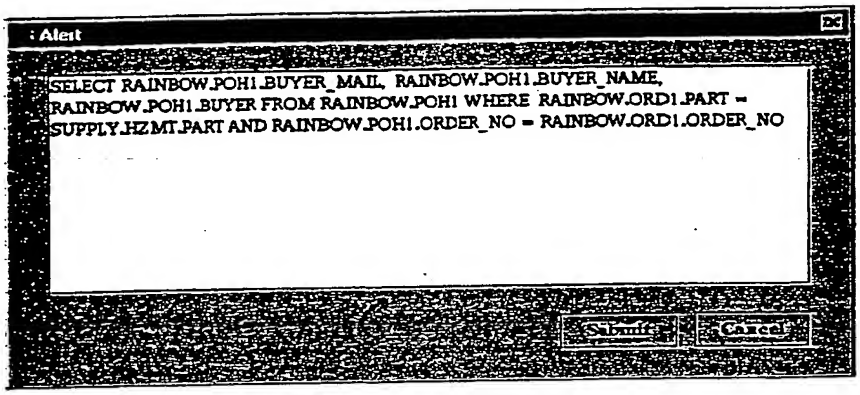

: Alert
SELECT RAINBOW.POH1.BUYER_MAIL, RAINBOW.POH1.BUYER_NAME,
RAINBOW.POH1.BUYER FROM RAINBOW.POH1 WHERE RAINBOW.ORD1.PART =
SUPPLY.HZMT.PART AND RAINBOW.POH1.ORDER_NO = RAINBOW.ORD1.ORDER_NO
Submit
Cancel

FIG. 7

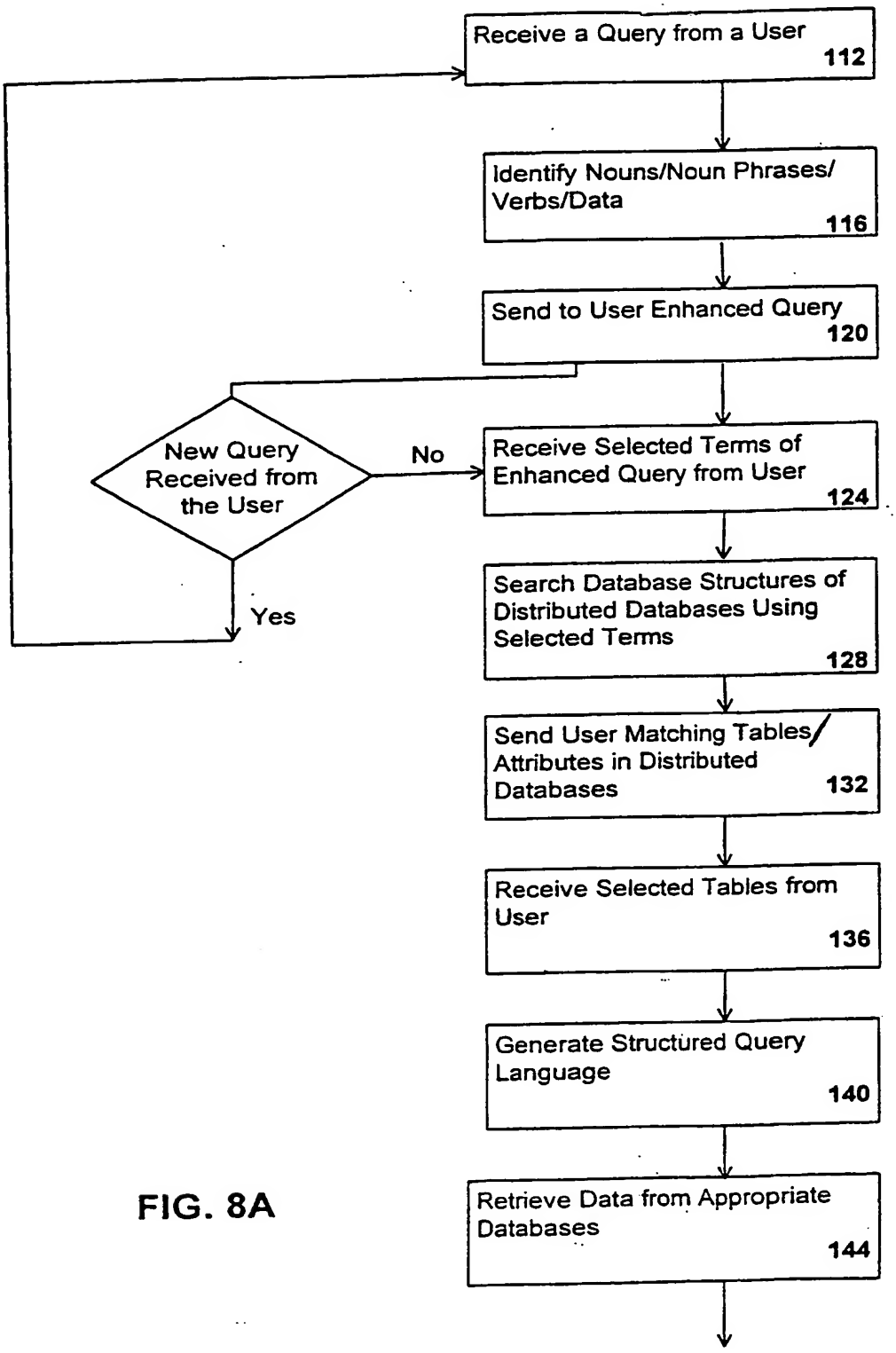

Receive a Query from a User
112
Identify Nouns/Noun Phrases/ Verbs/Data
116
Send to User Enhanced Query
120
New Query Received from the User
No
Yes
Receive Selected Terms of Enhanced Query from User
124
Search Database Structures of Distributed Databases Using Selected Terms
128
Send User Matching Tables/ Attributes in Distributed Databases
132
Receive Selected Tables from User
136
Generate Structured Query Language
140
Retrieve Data from Appropriate Databases
144

FIG. 8A

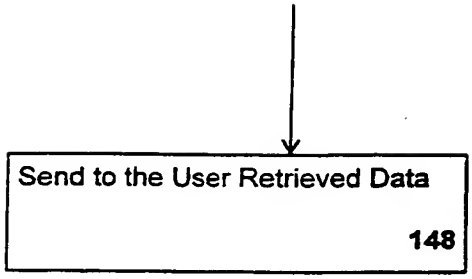
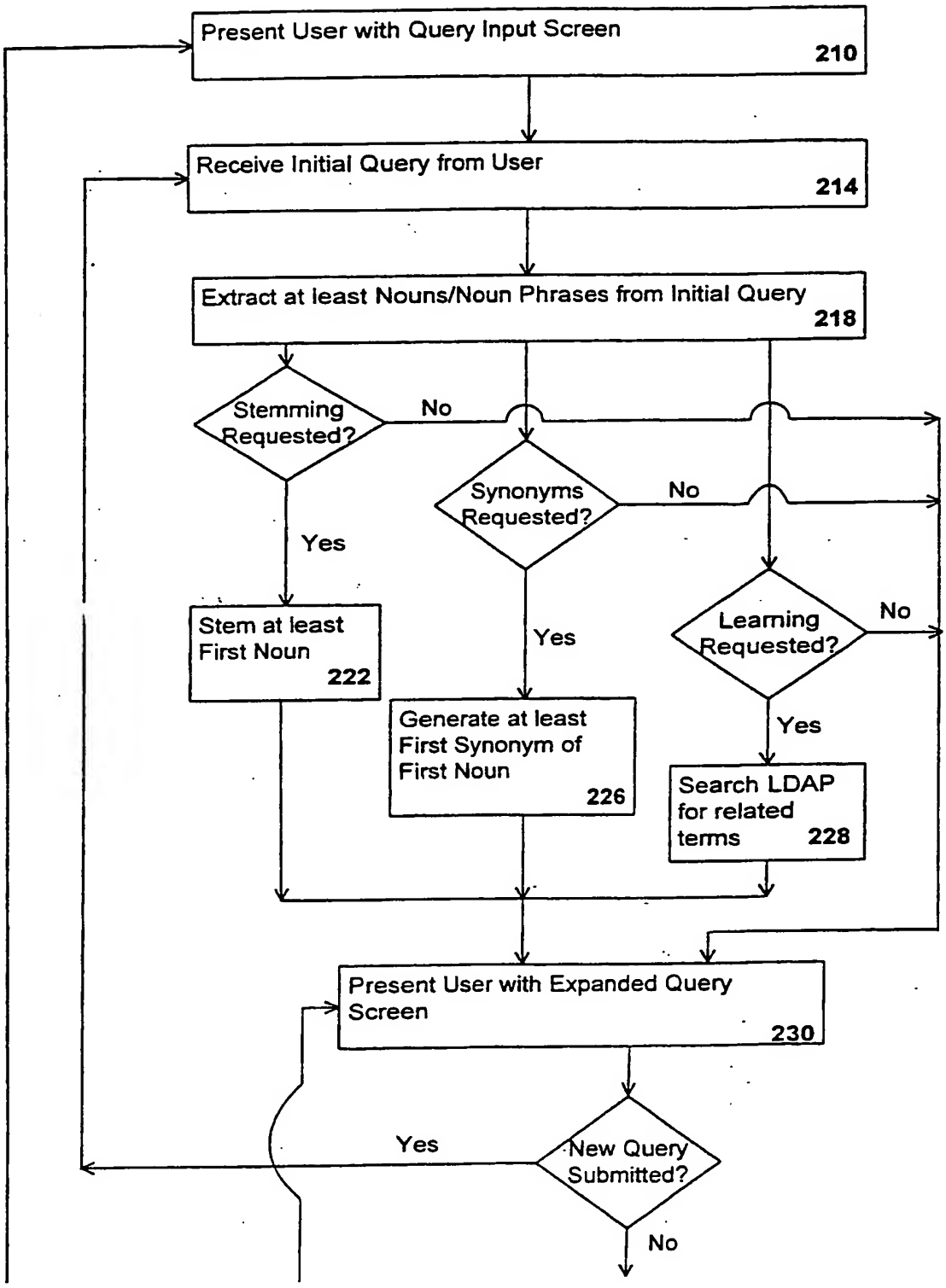



FIG. 8B

FIG. 9A


Present User with Query Input Screen
210
Receive Initial Query from User
214
Extract at least Nouns/Noun Phrases from Initial Query
218
Stemming Requested?
No
Yes
Synonyms Requested?
No
Yes
Learning Requested?
No
Yes
Stem at least First Noun
222
Generate at least First Synonym of First Noun
226
Search LDAP for related terms
228
Present User with Expanded Query Screen
230
New Query Submitted?
Yes
No

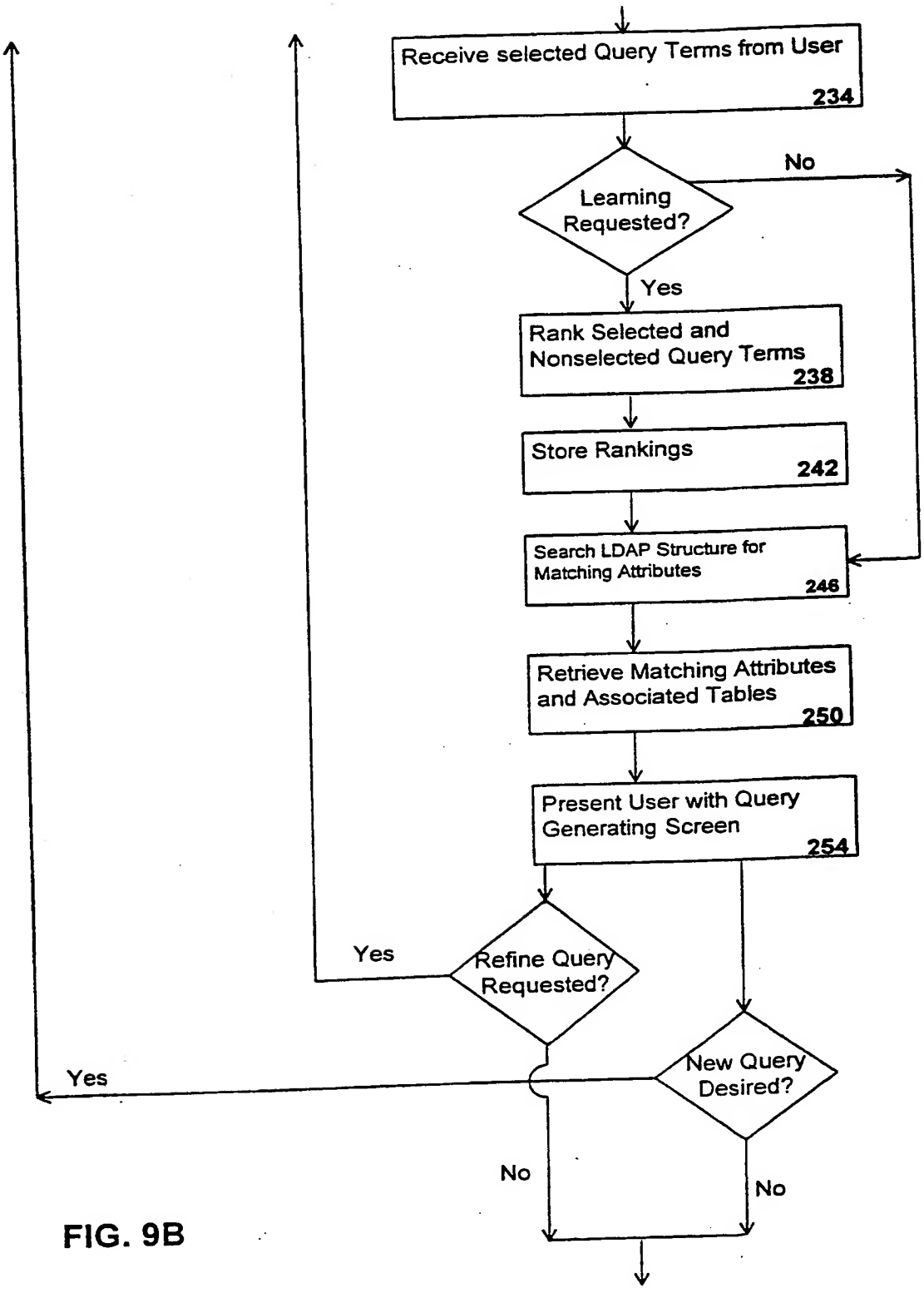
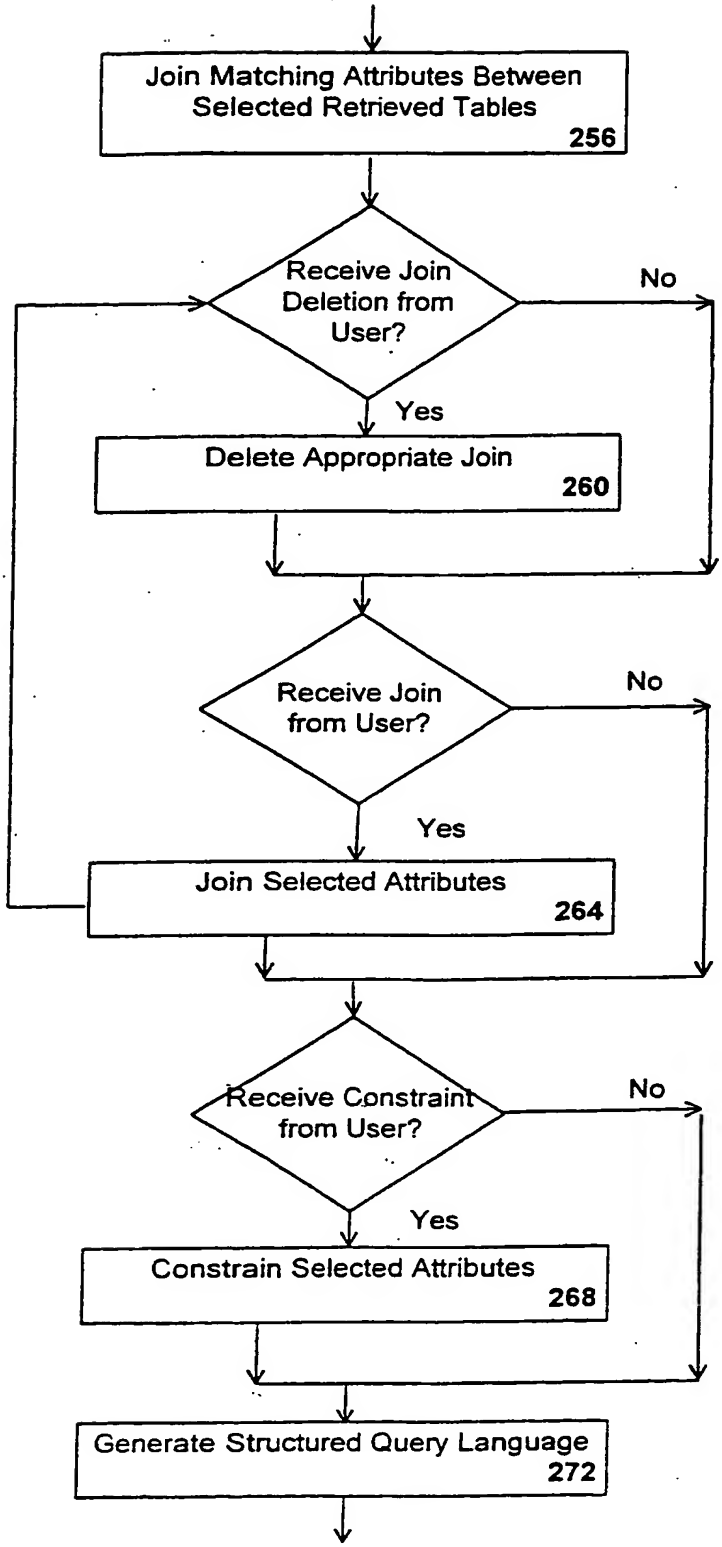

Receive selected Query Terms from User
234
Learning Requested?
No
Yes
Rank Selected and Nonselected Query Terms
238
Store Rankings
242
Search LDAP Structure for Matching Attributes
246
Retrieve Matching Attributes and Associated Tables
250
Present User with Query Generating Screen
254
Refine Query Requested?
Yes
New Query Desired?
Yes
No
No

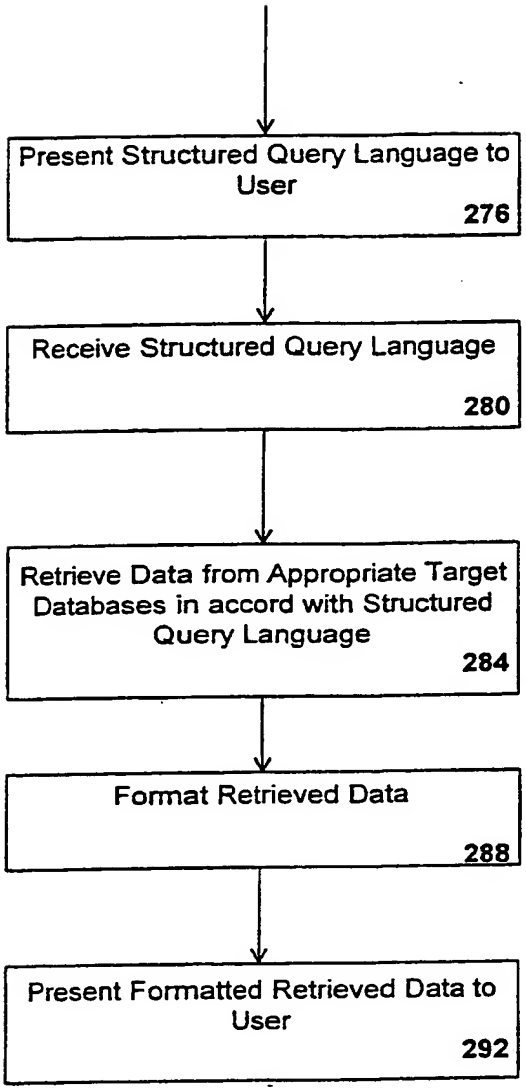
FIG. 9B

FIG. 9C




RNSDOCID: <WO 0141002A1 I >

FIG. 9D


Present Structured Query Language to User
276
Receive Structured Query Language
280
Retrieve Data from Appropriate Target Databases in accord with Structured Query Language
284
Format Retrieved Data
288
Present Formatted Retrieved Data to User
292

# INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/32815

**A. CLASSIFICATION OF SUBJECT MATTER**

IPC(7) : G06F 17/30
US CL : 707/5, 4

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 707/1-5, 10

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EAST

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category * | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X | US 5,893,087 A (WLASCHIN et al) 06 April 1999 (03.04.1999), the abstract, Figures 2, 3, 11, 12, columns 2-3. | 7-13 |
| X | US 5,933,822 A (BRADEN-HARDER et al) 03 August 1999 (03.08.1999), the abstract, Figures 2, 4, columns 5-6. | 7-13 |

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
12 March 2001 (12.03.2001)

Date of mailing of the international search report
06 APR 2001

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703)305-3230

Authorized officer
Thomas Black
Telephone No. 305-9000

Form PCT/ISA/210 (second sheet) (July 1998)

THIS PAGE BLANK (USPTO)

**This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record**

## BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☑ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☑ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

**IMAGES ARE BEST AVAILABLE COPY.**
**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

THIS PAGE BLANK (USPTO)